

IN THE SPECIFICATION

Please replace the title with the following new title:

METHOD AND APPARATUS FOR BACKING-UP GAME DATA.

Page 1, line 4, delete the heading "BACKGROUND OF THE INVENTION".

Kindly enter the following paragraphs:

Page 1, replace the first paragraph starting at line 6 with the following:

The present invention relates to a game machine, a game backup control program and a game backup control method [[of]] for backing-up game data. More specifically, the present invention relates to a game machine provided with a nonvolatile memory having two or three more backup areas within an electrically rewritable storing area and game data is written into the backup area, and a backup control program and a backup control method [[of]] for the game data.

Page 1, replace the second paragraph starting at line 12 with the following:

Description of the prior art BACKGROUND AND SUMMARY OF THE INVENTION

An example of this kind of a conventional game machine is disclosed in a Japanese Laid-open Patent Laying-open No. 5-12139 [G06F 12/16, A63F 9/22, G06F 1/00, G06F 9/06, G11C 5/00] laid-open published on January 22, 1993. In a hot start system of the prior art, a backup portion is selected among storing areas for game 1 to game 3, provided in a static RAM of a game cartridge, in response to an operation by an operator. If no game data has been backed-up in a selected backup portion, a backup-objective data area of the scratch RAM provided in a video game machine is initialized so as to start a game at the beginning. On the other hand, if the game data has been backed-up in the selected backup portion, the game data is read into the scratch RAM of the video game machine so as to resume the game. As the game progresses, the game data in the scratch RAM is renewed, and for example, every time

of making a hero act, the backup-objective data out of the game data of the scratch RAM is automatically stored in a ~~determined~~ predetermined (selected) storing area in the static RAM of the game cartridge, whereby the game status can be automatically backed-up at any time.

Page 2, replace the first paragraph starting at line 2 with the following:

Furthermore, another example of the prior art is disclosed in a Japanese Patent Laying- Laid-open Patent Publication No.10-31611, [G06F 12/00, G11C 16/06] ~~laid-open published~~ on February 3, 1998. In a file system for a nonvolatile memory storage medium disclosed in [[of]] this prior art, a logical block number obtained by searching a minimum writing times information value among unassigned state information in a usage frequency list is made as a write-objective block[[,]] and whereby allowing, available memory blocks [[can]] to be written uniformly, that is, the frequencies of the writing to the blocks are made even so as to ~~intend to~~ effectively lengthen the life of the nonvolatile memory.

Page 2, replace the second paragraph starting at line 10 with the following:

However, in the former, [[the]] a static RAM is utilized as a backup memory backing-up the game data, and therefore, it is necessary to utilize a battery for ~~backing-up the game data~~. Thus, there is a [[high]] significant possibility that the battery is ~~dead in the game~~ will die during a game in which the battery is ~~especially exhausted low~~, and in such a case, there is a problem that all the game data within the static RAM may be lost. Recently, in order to avoid such ~~the problem, a problems~~, nonvolatile memory (e.g., [[flush]] flash memory, ferroelectric memory, [[and]] etc.), which needs not does not need a battery for holding data, ~~starts to be~~ is being utilized as a backup memory. Thus, with utilizing the nonvolatile memory, there is no fear of battery exhaustion. However, ~~especially, the flush~~ flash memory, has a disadvantage of being slow in writing speed and short in life ~~of storing elements~~ as compared with [[the]] static RAM. Therefore, if a backup portion selected by the user is ~~convergently~~ continually being rewritten, the life of the storing elements at the in that backup [[area]] area/portion of memory becomes shorter than another backup area/portion, and therefore, ~~there occurs~~

another problem occurs in that the game data stored in that continually rewritten backup portion of memory is apt to be lost.

Page 2, replace the third paragraph starting at line 23 with the following:

On the other hand, in the latter example, the number of rewriting times of each memory block of the nonvolatile memory is made even, thereby the same so as to prevent a specific memory block from being extremely shorter in life than other memory blocks.

Page 3, replace the first paragraph starting at line 1 with the following:

However, even in view of the above-described prior arts, there still exists problems that if a storing element (or a part of storing area) becomes abruptly defective (due to a short life and etc.) during the time the last game data is being written in the backup portion, it becomes impossible to write the last game data, and furthermore, due to a fact in the instance that a part of the last game data has been written onto older game data, the older game data may also be lost. Furthermore, in a case that [[a]] the power [[of]] to the game machine is turned-off [[at a]] midway during [[of]] writing the last game data, there occurs a problem that both the last game data and the older game data are lost, or [[and]] worse yet, all the game data is lost. It is noted that lost, due to damage, of the preventing/loss of game data obtained by progressing [[the]] through a game for long hours is critical for the player.

Page 3, replace the second paragraph starting at line 12 with the following:

Notable aspects of the illustrative exemplary implementations disclosed herein include: Therefore, it is a primary object of the present invention to provide a novel game machine apparatus, a game data backup control program and a backup control method.

Page 3, replace the third paragraph starting at line 15 with the following:

Another object of the present invention is to provide a game machine, a game data backup control program and a backup control method capable of surely reliably

preventing game data from at least one generation ago the previous game play from being lost.

Page 3, replace the fourth paragraph starting at line 18 with the following:

[[A]] An illustrative exemplary non-limiting implementation of a game machine according to the invention is provided with comprising a nonvolatile memory which has two or three [[more]] additional backup areas in an electrically rewritable storing area and which writes game data in [[the]] these backup areas. The game machine comprises an area selecting means for preferentially selecting, [[when]] whenever last game data is [[to be]] about to be written, a backup area having an older writing time among two or three more backup areas as a write -objective backup area [[of]] for the last game data; a writing means for executing a writing of the last game data into the write-objective backup area selected by the area selecting means; a determining means for determining whether or not the writing of the last game data could be performed by the writing means; a repeating means for repeating, when it is determined the writing of the last game data could not be performed by the determining means, a selection of the write-objective backup area by the area selecting means as necessary; and a prohibiting means for prohibiting, when only a backup area stored with game data written immediately before the last game data finally becomes a selectable state by the area selecting means, a writing to the write-objective backup area.

Page 4, replace the second paragraph starting at line 7 with the following:

Specifically, the game machine (10: reference numeral corresponding to a preferable embodiment described later and so forth) is provided with the nonvolatile memory (58) having at least the two backup areas in the electrically rewritable storing area, and the game data is written into the backup area. The area selecting means (40, S15) preferentially selects, when the last game data is to be written, the backup area stored with the game data having an older writing time among at least two backup areas as the write-objective backup area of the last game data. The writing means (40, S17, S25) executes the writing of the last game data into the write-objective backup area selected by the area selecting means (40, S15). The determining means (40, S27)

determines whether or not the writing of the last game data could be performed by the writing means (40, S17, S25). The repeating means (40, S101, S103) repeats, [[when]] whenever it is determined that the writing of the last game data could not be performed by the determining means (40, S27), the selection of the write-objective backup area by the area selecting means (40, S15) as necessary. More specifically, in [[a]] the case where the number of the backup areas is only two, even if it is determined that the last game data could not be written, the selection of the write-objective backup area by the area selecting means (40, S15) is not repeated. On the other hand, in [[a]] the case where the number of the backup areas is three or more, when it is determined the last game data could not be written, the selection of the write-objective backup area by the area selecting means (40, S15) is repeated so as to repeatedly execute the writing process of the last game data. Then, the prohibiting means (40, S29) prohibits, when only the backup areas stored with the game data written immediately before the last game data finally becomes the selectable state by the area selecting means (40, S15), the writing to the write-objective backup area.

Page 5, replace the second paragraph starting at line 5 with the following:

It is noted that in a [[case]] situation where the game data has not [[yet]] previously been stored in the backup area, the backup area is selected with [[a]] priority as a write-objective backup area.

Page 5, replace the third paragraph starting at line 7 with the following:

One additional feature of the non-limiting example implementation disclosed herein is that it makes it According to the present invention, it is possible to surely leave securely leave stored in memory the game data saved from [[one]] a previous generation [[ago]] of game play.

Page 5, replace the fourth paragraph starting at line 9 with the following:

In one embodiment one non-limiting example implementation of the game machine disclosed herein, the writing means writes historical data for discriminating between oldness and newness of the game data by being included in the last game

data, and the area selecting means selects, before writing the last game data, a backup area stored with game data written earlier than the last game data as the write-objective backup area on the basis of the historical data. More specifically, since the writing means (40, S17, S25) writes the last game data including the historical data for discriminating between the oldness and the newness of the game data, the area selecting means (40, S15) can select, before writing the last game data, the backup area stored with the game data having an older writing time as the write-objective backup area on the basis of the historical data included in the old data. Accordingly, for example, game data having the oldest historical data is determined, and whereby allowing, the backup area in which the oldest game data is stored [[can]] to be selected as the write-objective area.

Page 5, replace the fifth paragraph starting at line 21 with the following:

In another non-limiting example implementation of the game machine disclosed herein embodiment, the prohibiting means includes a ~~forcedly terminating~~ means for forcedly terminating a writing process of the last game data when only the backup area stored with the game data written immediately before the last game data finally becomes [[a]] selectable [[state]] as the write objective backup area by the area selecting means. More specifically, ~~the forcedly a terminating~~ means (40, S29) forcedly terminates the writing process of the last game data [[when]] whenever only the backup area stored with the game data written immediately before the last game data finally becomes selectable as [[the]] a write-objective backup area by the area selecting means, and therefore . Consequently, by this means, it is possible to surely securely leave the old game data [[onto]] saved in an area in memory for which overwriting is prohibited prevented. At this time, the last game data is canceled.

Page 6, replace the second paragraph starting at line 6 with the following:

In yet another non-limiting example implementation of a game machine disclosed herein, The other embodiment further comprises a message displaying means for displaying a predetermined alarm message when the is displayed whenever overwriting of game data is prohibited by the prohibiting means. More specifically, when the writing

overwriting is prohibited by the prohibiting means (40, S29), a [[the]] message displaying means (14, 40, S31) displays [[the]] a predetermined alarm message so as to urge the user to exchange the nonvolatile memory.

Page 6, replace the third paragraph starting at line 11 with the following:

The backup control program of the game data according to the present invention is One aspect of the exemplary implementation disclosed herein comprises a backup control program by arrangements in which the last game data is written [[in]] into two or [[three]] more backup areas in an electrically rewritable ~~storing storage~~ area of a nonvolatile memory connected to [[a]] the game machine. [[The]] In this example implementation, the backup control arrangement program makes a computer (processor) of the game machine execute following steps of perform: an area selecting step selection process that, whenever for preferentially selecting, when last game data is [[to be]] written, selects a backup area containing stored [[with]] game data having of an older writing time among two or [[three]] more backup areas as a write-objective backup area of for writing (saving) the last game data; a writing step for executing a writing of the last game data to the selected write-objective backup area selected by the area selecting step; a determining step for determining whether or not the writing of the last game data could be performed by the writing step; a repeating step for repeating the selection of the write-objective backup area as necessary, [[when]] if it is determined that the writing of the last game data could not be performed by the determining step, a selection of the write-objective backup area by the area selecting means as necessary; and, a prohibiting step for prohibiting a writing to the write-objective backup area, when [[only]] a backup area stored with game data written immediately before the last game data finally becomes [[a]] selectable [[state]] by the area selecting step, a writing to the write-objective backup area.

Page 7, replace the first paragraph starting at line 1 with the following:

Another aspect of the exemplary implementation disclosed herein includes A backup control method according to the present invention is a backup control method by which game data is written in two or three more backup areas in an electrically

rewritable storing area of a nonvolatile memory connected to a game machine, and includes following which comprises steps of: (a) selecting, as a write-objective backup area a backup area, which stores game data having an older writing time and to which the last game data can be written[[,]]; (b) canceling the writing of last game data, when a writing into the write-objective backup area is not executable and only a backup area that was stored with game data immediately before the last game data is selectable as a write-objective backup area, so as to leave intact the game data that was written immediately before the last game data.

Page 7, replace the second paragraph starting at line 10 with the following:

In such In the presently disclosed non-limiting example implementations of both the game data backup control program apparatus and the backup control method [[also]], it [[is]] becomes possible to surely securely leave [[the]] game data from a previous [[one]] generation [[ago]] of game play in memory, as is similarly the case similarly to the for the above-described example game machine implementation.

Page 7, replace the third paragraph starting at line 13 with the following:

The above described objects and other objects, non-limiting features, aspects and advantages of the present invention example implementations described herein will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

Page 7, replace the fourth paragraph starting at line 18 with the following:

Figure 1 is an illustrative view showing one example implementation of an appearance of a game machine of the present invention;

Page 7, replace the fifth paragraph starting at line 20 with the following:

Figure 2 is a block view showing an electrical configuration of the game machine shown in Figure 1 embodiment;

Page 7, replace the seventh paragraph starting at line 24 with the following:

Figure 4 is an illustrative view showing a selection of game data backup areas in an example implementation wherein a case there is provided with two backup areas in a [[flush]] flash memory as shown in Figure 2 is provided with two backup areas;

Page 8, replace the first paragraph starting at line 1 with the following:

Figure 5 is an illustrative view showing a selection of backup areas in a case there for an example wherein is provided with two backup areas in the flush a flash memory as shown in Figure 2 is provided with two backup areas;

Page 8, replace the second paragraph starting at line 3 with the following:

Figure 6 is an illustrative view showing [[a]] an example state in which in-a-case the two backup areas of the flush a flash memory shown in as illustrated by Figure 2 are provided, wherein when one backup area becomes unwritable, and the other backup area is prohibited from being written;

Page 8, replace the third paragraph starting at line 6 with the following:

Figure 7 is an illustrative view showing one example of a save screen, a save end screen and an error screen as displayed on an LCD display device as shown in Figure 1 and Figure 2;

Page 8, replace the fourth paragraph starting at line 8 with the following:

Figure 8 is an illustrative view showing a state in which in-a-case the two backup areas of the flush a flash memory shown in as illustrated by Figure 2 are provided and a writing prohibiting flag area is further provided, wherein when one backup area becomes unwritable, and the other backup area is prohibited from being written;

Page 8, replace the fifth paragraph starting at line 12 with the following:

Figure 9 is a flowchart showing one example implementation of an overall process [[of]] performed by a CPU shown in Figure 2;

Page 8, replace the sixth paragraph starting at line 14 with the following:

Figure 10 is a flowchart showing [[a]] an example backup process [[of]] performed by the CPU shown in Figure 2;

Page 8, replace the seventh paragraph starting at line 15 with the following:

Figure 11 is a flowchart showing [[a]] an example selection process [[of]] performed by the backup areas of the CPU shown in Figure 2;

Page 8, replace the eighth paragraph starting at line 17 with the following:

Figure 12 is a flowchart showing [[a]] an example renewal process [[of]] performed by historical data of the CPU shown in Figure 2;

Page 8, replace the ninth paragraph starting at line 19 with the following:

Figure 13 is an illustrative view showing one example of a memory map of a [[flush]] flash memory provided in a game cartridge loaded in another game machine ~~of another embodiment of the present invention~~;

Page 8, replace the tenth paragraph starting at line 22 with the following:

Figure 14 is a flowchart showing a part of a backup process [[of a]] performed by the CPU shown in Figure 13 ~~embodiment~~;

Page 8, replace the eleventh paragraph starting at line 24 with the following:

Figure 15 is a flowchart showing another part of the backup process [[of]] performed by the CPU shown in Figure 13 ~~embodiment~~; and

Page 9, replace the first paragraph starting at line 1 with the following:

Figure 16 is an illustrative view showing another example of the memory map of the [[flush]] flash memory provided in the game cartridge loaded into the example game machine [[of]] illustrated in Figure 13 embodiment.

Page 9, replace the second paragraph starting at line 6 with the following:

Referring to Figure 1, a game machine 10 of this embodiment one example implementation includes a case 12. The case 12 is, on the surface thereof, provided with a color liquid crystal display 14 (hereinafter, referred to as “LCD”) at an approximately center. The LCD 14 is displayed with a game space and game characters appearing in the game space, and messages as necessary. The case 12 is on its surface provided with operating buttons 16, 18, 20, 22, 24, 26 and 28. The operating buttons 16, 18 and 20 are placed at a left of the LCD 14, and the operating buttons 22 and 24 are placed at a right of the LCD 14. Furthermore, the operating buttons 26 and 28 are placed at an upper end surface (above the LCD) of the case 12.

Page 11, replace the third paragraph starting at line 12 with the following:

Furthermore, the game cartridge 32 is provided with a ROM 56 and a [[flush]] flash memory 58 of one example of a nonvolatile memory, and although not illustrated, the ROM 56 and the [[flush]] flash memory 58 are connected with each other via a bus and connected to connectors. Accordingly, as described above, when the game cartridge 32 is loaded into the game machine 10, the CPU 40 is electrically connected to the ROM 56 and the [[flush]] flash memory 58.

Page 11, replace the fourth paragraph starting at line 18 with the following:

It is noted that although the [[flush]] flash memory 58 is utilized in this embodiment, as another nonvolatile memory, a ferroelectric memory (FeRAM), EEPROM and etc. can also be utilized.

Page 12, replace the second paragraph starting at line 5 with the following:

In addition, the [[flush]] flash memory 58 is stored with game data (backup data). For example, when a user plays the game by utilizing the game machine 10, game data is stored (renewed) in the work memory 44 of the game machine 10 in accordance with the progress of the game. Then, the game data is written (saved) into the [[flush]] flash memory 58 of the game cartridge 32 in response to an instruction of the user or at a predetermined timing (event). Thereafter, in a case the game is to be continued, the game is advanced as it is, and then, storing and backup of the game data described

above are executed. Furthermore, in a case of ending the game once, the user reads the game data backed-up the last time and starts or resumes next time where the user left off.

Page 12, replace the third paragraph starting at line 14 with the following:

~~Herein, the flush~~ The flash memory 58, as compared with a memory such as SRAM, has an advantage of requiring no backup power source but has disadvantage of being short in rewritable life and in rewriting speed. Especially, at a time of writing data, when unwritable problems such as physical breakdown, electrical breakdown and etc. occur, a part of the game data cannot be written and hence, breakage of the game data occurs. Especially, the game data is frequently rewritten as the game progresses as described above, and therefore, without taking measures, the breakage of the game data clearly exists.

Page 13, replace the first paragraph starting at line 1 with the following:

~~Herein, in this embodiment~~ In one example implementation, the [[flush]] flash memory 58 is provided with two backup areas in which the game data is sequentially stored. Thus, even if one backup area is broken down, the other backup area surely retains the game data, and whereby, it is possible to prevent all the data from being completely lost. Furthermore, when only thing to be done is to perform overwrite to game data (old game data) written immediately before, overwriting on the old data written immediately before is prohibited. Thus, a loss of the game data due to a trouble occurring during overwriting the last game data onto the old game data can be prevented.

Page 13, replace the second paragraph starting at line 9 with the following:

It is noted that the [[flush]] flash memory 58 generally needs to be rewritten with data sector (block) by sector (block), but some products are rewritable address by address, and in this embodiment, the backup area includes one or two more sectors. That is, the number of the sectors included in the backup area is determined depending upon a volume of the game data.

Page 13, replace the third paragraph starting at line 14 with the following:

The [[flush]] flash memory 58 has a first backup area 58a and a second backup area 58b as shown in Figure 4 (A). Furthermore, as described above, the game data includes (is added with) historical data so as to sequentially write the game data to the first backup area 58a and the second backup area 58b in this embodiment. That is, Figure 4 (A) shows a state that game data having historical data (1) is written to the first backup area 58a, and game data having historical data (2) is written to the second backup area 58b.

Page 13, replace the fifth paragraph starting at line 23 with the following:

Herein, in a case the game data (last game data) stored in the work memory 44 of the game machine 10 is written to the [[flush]] flash memory 58, the historical data (3) is assigned to the last game data, and an area to which the last game data is to be written is selected between the first backup area 58a and the second backup area 58b according to a predetermined rule (condition for area selection).

Page 14, replace the third paragraph starting at line 12 with the following:

Furthermore, the condition for area selection is represented by four inequalities of $2>1$, $3>2$, $4>3$, $1>4$. These numerals 1 to 4 are numerical values indicated by the historical data and mean that the game data including the historical data of a numeral on the right side is older than the game data including the historical data of a numeral on the left side (written to the [[flush]] flash memory 58 earlier). Such the condition for area selection is determined (defined) by inequalities (conditions) utilized in correspondence to the historical data of the last game data, and specifically represented by a table 1.

Page 16, replace the third paragraph starting at line 19 with the following:

Furthermore, in a case the game data cannot be written to the [[flush]] flash memory 58, an alarm message is displayed on the LCD 14 so as to urge the user to exchange the [[flush]] flash memory 58 (to repair the game cartridge 32). For example, when the user instructs to store the game data (display a save screen), in response

thereto a screen for selecting whether or not the saving is to be executed (save screen) is displayed on the LCD 14 as shown in Figure 7 (A). Specifically, the CPU 40 reads image data of the save screen from the image data 562 stored in the ROM 56 of the game cartridge 32 according to an instruction of the user and develops the same on the VRAM 50. Then, the LCD driver 52 reads the image data of the save screen developed on the VRAM 50 according to the instruction of the CPU 40 and outputs the same on the LCD 14.

Page 17, replace the fourth paragraph starting at line 19 with the following:

As described above, it is possible to surely securely leave the game data from at least one previous generation stored in the flash memory [[ago]], and furthermore, when the alarm message is displayed, e.g. (by sending the cartridge for repair to a manufacturer), ~~for example~~, it is possible to exchange with a new [[flush]] flash memory with the game data one generation ago left. This allows, although the game data is one generation ago, the player to resume where the player left off.

Page 17, replace the fifth paragraph starting at line 24 with the following:

It is noted that in a case that after one backup area becomes unwritable and the other backup area is prohibited to be written, the player further advances the game, a message indicative of an unwritable state is displayed at only a time that the last game data is to be stored, and therefore, there is a fear that a previous effort by the player comes to nothing. Therefore, as shown in Figure 8, a write prohibiting flag 58c is further provided within the [[flush]] memory 58, and a prohibiting flag of the backup area to which the overwriting is prohibited is set (turned-on), that is, overwriting in the backup area is prohibited, the alarm message can be quickly displayed with reference to the write prohibiting flag 58c at a time of the game start or at a time of being instructed to display the save screen, and therefore, it is possible to eliminate the above-described wasted effort.

Page 19, replace the fifth paragraph starting at line 25 with the following:

In a following step S17, the last data and the historical data are written to the

selected write-objective backup area. That is, a writing process of the last game data including the historical data is executed. In a step S19, it is determined whether or not a normal writing is performed by a checksum. That is, the CPU 40 reads the written game data from the [[flush]] flash memory 58 and compares the game data with the game data stored in the work memory 44 so as to determine if there is a match or coincidence. If there occurs a match, that is, if “YES” in the step S19, it is determined the normal writing is performed, a renewal process of the historical data is executed in a step S21, and then, a save end screen shown in Figure 7 (B) is displayed in a step S23 so as to return to the backup process.

Page 20, replace the fourth paragraph starting at line 24 with the following:

It is noted that in a case of providing the write prohibiting flag area 58c in the [[flush]] flash memory 58 as shown in Figure 8, it is appropriate that writing of the write prohibiting flag is executed when the writing of the game data is ended in the step S29, and then, the error screen is displayed in the step S31. This makes it possible to easily know whether or not the writing of the last game data is prohibited by referring with the write prohibiting flag area at a time of starting the game or at a time that the save screen is instructed to be displayed, and therefore, it is possible to immediately display the error screen.

Page 21, replace the fourth paragraph starting at line 19 with the following:

It is noted that in a case the game data is not stored in the [[flush]] flash memory 58 in an initial state, it is impossible to refer the historical data. Therefore, priorities are assigned to respective areas, and until the game data is written to all the backup areas, the game data may be written according to the priorities. That is, it is possible to write the game data from the first backup area 58a to the second backup area 58b in this order or in reverse order thereto.

Page 22, replace the third paragraph starting at line 8 with the following:

~~According to the embodiment~~ In at least one example implementation, when writing of the game data to one backup area becomes disabled ~~disabled~~, writing of the

game data to the other back-up area is prohibited, and therefore, it [[is]] becomes possible to surely securely leave [[the]] game data from one previous generation [[ago]] stored in memory. That is, it is possible to prevent the previously stored game data from being damaged.

Page 22, replace the fifth paragraph starting at line 15 with the following:

The game machine 10 of another embodiment example implementation, discussed below, is the same as the above-described embodiment implementation except that three or more backup areas are formed in the [[flush]] flash memory 58 provided in the game cartridge 32, and therefore, a duplicated description is omitted.

Page 22, replace the sixth paragraph starting at line 18 with the following:

As shown in Figure 13, the [[flush]] flash memory 58 is provided with N of the backup areas 58n ($1 \leq n$ (natural number) $\leq N$) ~~in this embodiment~~. In this embodiment also example implementation, the game data is sequentially stored in respective one of the backup areas 58n in the same manner as the above-described embodiment. For example, until the game data is stored in all the backup areas 58n, the backup areas are selected from $n=1$ in order so as to store the game data; however, after the game data is stored in all the backup areas 58n, the game data is overwritten according to a predetermined condition for area selection.

Page 24, replace the second paragraph starting at line 4 with the following:

In addition, similar to the above-described embodiment implementation, the game data including the historical data indicated by the numeral on the right side of the inequality is older than the game data including the historical data indicated by the numeral on the left side of the inequality. Accordingly, in a case of writing the last game data according to the condition for area selection, the backup area stored with the game data including the historical data of the numeral described at the right end (far-right side) is selected as the write-objective backup area.

Page 24, replace the third paragraph starting at line 11, with the following:

It is noted that in a case the backup area which becomes unwritable exists, the historical data of the game data stored in the backup areas except for the backup area which becomes unwritable is acquired, and by applying them to the condition for area selection, the write-objective backup area is selected. Accordingly, in a case the backup area stored with the game data including the historical data having a numeral at the far-right side of the condition for area selection becomes unwritable, the backup area stored with next older game data (game data including the historical data having a numeral at a second position from the right end) is selected as the write-objective area. That is, the read historical data is compared according to the condition for area selection, and consequently, the backup area stored with the game data including the historical data of the numeral at the far-right side is selected as the write-objective backup area. In other words, the backup area stored with the game data having an older time of writing to the [[flush]] flash memory 58 is preferentially selected.

Page 24, replace the fourth paragraph starting at line 24 with the following:

Next, a description is ~~made on a detailed~~ provided of the process ~~of the~~ implemented by CPU 40; however, since the process is the same as the [[above]] previously-described embodiment implementation except that a part of the backup process is changed, and so therefore, a ~~duplicated redundant~~ description of the process is omitted.

Page 26, replace the second paragraph starting at line 11 with the following:

Furthermore, in a case where the unwritable flag area 58p is not provided, if data indicative of whether writable or unwritable is written to the end cell of each backup area, as in the above described-embodiment previous example implementation, it is possible to know the unwritable backup area 58n.

Page 26, replace the third paragraph starting at line 15 with the following:

According to this embodiment example implementation, when writing is

performed to the oldest game data and writing of the game data is disabled with respect to all the backup areas but one, writing of the game data to the one backup area is prohibited, and therefore, it is possible to surely leave the game data one generation ago. That is, it is possible to prevent the game data from being damaged.